Lecture Notes in Integer Programming*

Andrea Lodi †

Version 1.2, December 11, 2013

Abstract

These are the lecture notes for the course of *Optimization Models and Algorithms M*, academic year 2013-2014. In particular, the notes cover *(i)* the modeling of basic problems by means of (Mixed-)Integer Linear Programming and discussing what a *good* model is, and *(ii)* the algorithms for models with exponentially-many rows or columns.

Contents

*Notes heavily based on the notes in Italian by my friend and colleague Alberto Caprara (1968-2012)

†andrea.lodi@unibo.it. DEI, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

1 ILP Models for "clean" NP-hard problems

This section covers the part of the course devoted to modeling basic problems by means of (Mixed-)Integer Linear Programming, (M)ILP, and discussing what a *good* model is.

1.1 Uncapacitated Facility Location

We are given

- m *clients* to be served,
- n *facilities* (or service centers) that can be opened or not,
- for each facility j, f_j is the *cost of opening* facility j, and
- for each client i and each facility j, c_{ij} is the *cost of serving* client i by facility j

The so-called *Uncapacitated Facility Location* Problem (UFLP) calls for determining (i) which facilities need to be opened, and (ii) which one among the open facilities serves each client, in such a way that the overall cost, i.e., the sum of opening and service costs, is a minimum

It is easy to see that once the decision on which facilities must be opened, then each client i will be assigned to the open facility j such that the cost c_{ij} is a minimum. Thus, deciding on the open facilities is the key decision and is taken by using binary variables

$$y_j := \begin{cases} 1, & \text{if facility } j \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

Nevertheless, without specific variables that define the assignment of clients to facilities it is not possible to have a complete ILP model for UFLP, thus the following binary variables need to be introduced

$$x_{ij} := \begin{cases} 1, & \text{if client } i \text{ is assigned to facility } j \\ 0, & \text{otherwise} \end{cases}$$

By using the two sets of variables above one can write the following model

$$\min \sum_{j=1}^{n} f_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m \tag{2}$$

$$x_{ij} = 1 \Rightarrow y_j = 1, \quad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{3}$$

$$y_j, x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{4}$$

Of course, the model above is not an ILP model because logic constraints (3) must be expressed with linear equations or inequalities like

$$x_{ij} \le y_j, \quad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{5}$$

or

$$\sum_{i=1}^{m} x_{ij} \le m y_j, \quad j = 1, \ldots, n \tag{6}$$

Both constraints (5) and (6) alone suffice to define valid ILP models for the UFLP. Thus, one would be tempted to use constraints (6) because they are way less (n versus nm).

However, constraints (5) and (6) are not equivalent in terms of LP relaxation. Indeed, constraints (6) are obtained by summing constraints (5) for $i = 1, \ldots, m$, thus any feasible LP solution of the model using constraints (5) is actually feasible for the model using constraints (6) but vice versa does not hold as shown by the following example

Example 1 Consider the special case with $n = m$, $f_j = 1$ and $c_{jj} = 0$ for $j = 1, \ldots, n$, $c_{ij} = +\infty$ for $i \neq j$

The optimal solution of UFLP coincides with the optimal solution of its LP relaxation with constraints (5), is given by $y_j = x_{jj} = 1$ for $j = 1, \ldots, n$, and has value n

The optimal solution of the LP (continuous) relaxation of the ILP model using constraints (6) is instead $y_j = 1/n$, $x_{jj} = 1$ for $j = 1, \ldots, n$, and has value 1 $\square$

Example 1 shows that the UFLP formulation using constraints (5) *dominates* that using constraints (6)

Finally, observe that, for both formulations, the constraints forcing the x variables being binary are *redundant* because for each integer y, it is always convenient to set $x_{ij} = 1$ for j such that $c_{ij} = \min\{c_{ik} : y_k = 1\}$

1.2 Set Covering, Partitioning and Packing

One of the most famous and important problems in combinatorial optimization, the so-called *Set Covering* Problem (SCP), can be seen as a special case of UFLP in which all entries of matrix c are either 0 or $+\infty$. In other words, that is the special case in which either client i can be served by facility j at null cost, or it *cannot* be served at all

The corresponding ILP formulation is significantly simpler because SCP "simply" calls for determining the subset of facilities to be opened in such a way that (i) the overall opening cost is a minimum, and (ii) all clients can be served

For each client i, let

$$J_i := \{j : c_{ij} = 0\}, \quad i = 1, \dots, m$$

be the the set of facilities that can serve client i. Then, the following ILP model with only y suffices

$$\min \sum_{j=1}^{n} f_j y_j \tag{7}$$

$$\sum_{j \in J_i} y_j \geq 1, \quad i = 1, \dots, m \tag{8}$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, n \tag{9}$$

The ILP model (7)–(9) defines a *generic* ILP in which

- all variables are binary,
- all constraints are inequalities in the form "$\geq$",
- all right hand sides are equal to 1, and
- the entries of the constraint matrix A are binary

Thus, the compact formulation of SCP is

$$\begin{aligned} \min c^T x & \\ Ax &\geq \mathbf{1} \\ x &\in \{0,1\}^n \end{aligned} \tag{10}$$

where $A \in \{0, 1\}^{m \times n}$, and $\mathbf{1}$ is the all-1 vector of m elements

There are two variants of SCP extremely important both in theory and in practice. The first, the so-called *Set Partitioning* Problem, is obtained by replacing inequalities with equations

$$\begin{aligned} \min c^T x & \\ Ax &= \mathbf{1} \\ x &\in \{0,1\}^n \end{aligned} \tag{11}$$

The second, the so-called *Set Packing* Problem, is obtained when the inequalities in the form "$\geq$" are replaced by inequalities in the form "$\leq$", thus naturally leading to express the objective function in maximization form

$$\begin{aligned} \max c^T x & \\ Ax &\leq \mathbf{1} \\ x &\in \{0,1\}^n \end{aligned} \tag{12}$$

It is easy to observe that the Set Packing problem always admits a feasible solution $x = (0, \ldots, 0)$, that the Set Covering problem admits a feasible solution if and only if $x = (1, \ldots, 1)$ is feasible, while one can prove that deciding if a feasible solution of the Set Partitioning problem exists is NP-complete

Although very similar in terms of formulation, the three problems are actually very different

- On the practical side, the Set Covering is less "difficult" and, generally, its LP relaxation is "strong" (although not easy to strengthen further)
- Set Packing has a direct interpretation as a graph problem (as discussed in the following) that indicates a clear way of strengthening its LP relaxation that, otherwise, is generally weak

Finally, it is easy to see that requiring $x_j \in \{0, 1\}$ or "$x_j \geq 0$, integer" is equivalent for all problems. In other words, the upper bound $x_j \leq 1$ in the LP relaxation is *redundant*

1.3 Capacitated Facility Location

The *Capacitated Facility Location* problem (CFLP) is the variant of UFLP in which

- each client i has an associated *demand* d_i, and
- each facility j has a *capacity* b_j, corresponding to the overall quantity of demand that can satisfy

The associated ILP model obtained by using constraints (5) of the UFLP is

$$\min \sum_{j=1}^{n} f_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{13}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \dots, m \tag{14}$$

$$x_{ij} \leq y_j, \quad i = 1, \dots, m, \ j = 1, \dots, n \tag{15}$$

$$\sum_{i=1}^{m} d_i x_{ij} \leq b_j y_j, \quad j = 1, \dots, n \tag{16}$$

$$y_j, x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \ j = 1, \dots, n \tag{17}$$

Observe that, differently from UFLP, the integer requirements for variables x are now necessary (unless serving the demand of a client by using multiple facilities makes sense from a practical perspective)

Finally, constraints (15) are not necessary for stating the validity of the model but they generally strengthen its LP relaxation

1.4 Bin Packing and Knapsack

Another very important problem in combinatorial optimization is the so-called *Bin Packing* Problem (BPP) that can be stated as the special case of CFLP in which

- service costs are all null: $c_{ij} = 0,\ i = 1, \ldots, m,\ j = 1, \ldots, n$, and
- opening costs and capacities are equal for all facilities: $f_j = 1,\ b_j = b,\ j = 1, \ldots, n$

Thus, all facilities are *identical* and BPP calls for opening the *minimum number* of facilities so as to serve all clients

A trivial modification of model (13)–(17) is as follows

$$\min \sum_{j=1}^{n} y_j \tag{18}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m \tag{19}$$

$$x_{ij} \leq y_j, \quad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{20}$$

$$\sum_{i=1}^{m} d_i x_{ij} \leq b y_j, \quad j = 1, \ldots, n \tag{21}$$

$$y_j, x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, m,\ j = 1, \ldots, n \tag{22}$$

A natural way of describing BPP is however different from the client/facility context. Namely, given m *items* with *weight* $d_1, \ldots, d_m$ and n (identical) *containers* (or *bins*) with *capacity* b, BPP calls for *packing* each item in a bin such that

- for each bin, the overall weight of the packed items does not exceed the capacity, and
- the number of used bins is a minimum

Without loss of generality we assume $\sum_{i=1}^{m} d_i > b$. Otherwise, the problem is trivial

BPP is one of the most basic and well-studied problems in the very large area of *Cutting & Packing*. An even more basic problem in that class is the so-called (0-1) *Knapsack* Problem (KP).

Formally, in KP each item i is characterized by a *profit* p_i ($i = 1, \ldots, m$) in addition to the weight w_i, a *unique* bin of capacity b is given, and we are asked to select a subset of the items that fit into the bin and whose overall profit is a maximum. The corresponding ILP model is as follows

$$
\begin{aligned}
\max \sum_{i=1}^{m} p_i x_i & & & \qquad (23)\\
\sum_{i=1}^{m} d_i x_i & \leq & b & \qquad (24)\\
x_i & \in & \{0,1\}, \quad i = 1,\ldots,m & \qquad (25)
\end{aligned}
$$

Although one can clearly use the Simplex algorithm to solve the LP relaxation of model (23)–(25) above, there is a much easier and faster algorithm to solve it *combinatorially*. The method due to Dantzig is reported in Algorithm 1

Algorithm 1 Solving the LP relaxation of Knapsack

sort the items according to $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_m}{w_m}$;
$\bar{b} := b$; // *residual* capacity
$x_i = 0$, $i = 1, \ldots, m$; // initialization
for $i = 1, \ldots, m$ **do**
  if $w_i \leq \bar{b}$ **then**
    $x_i := 1$;
    $\bar{b} := \bar{b} - w_i$
  else
    $x_i := \frac{\bar{b}}{w_i}$ // *critical* item
    return x
  end if
end for

Coming back to the BPP, a trivial lower bound on the number of bins required to pack all items is given by

$$\ell := \frac{\sum_{i=1}^{m} d_i}{b}$$

Unfortunately, ℓ is also the value of the following (feasible) solution of the LP relaxation of model (18)–(22)

$$y_j = \ell/n,\ j = 1,\ldots,n; \quad x_{ij} = 1/n,\ i = 1,\ldots,m,\ j = 1,\ldots,n$$

that in turn indicates that model (18)–(22) is weak because its optimal solution value is not better than a trivial solution like ℓ

Another severe drawback of using model (18)–(22) for solving BPP is its heavy *symmetry*: for each integer solution of value k, there exist $\binom{n}{k}k!$ equivalent solutions, with the practical effect that the branching constraints of a branch-and-bound algorithm are largely ineffective in solving the problem by enumeration

The two outlined drawbacks limit heavily the use of model (18)–(22) for solving BPP in practice. Although it is conceivable trying to strengthen it, it turns out to be more effective to define a completely new model for BPP

The alternative model contains an exponential (in m) number of variables because each variable corresponds to a feasible packing of items into a bin

Formally, let $\mathcal{S}'$ be the collection of all subsets of items that can be packed together in a bin without exceeding its capacity

$$\mathcal{S}' := \left\{ S \subseteq \{1, \dots, m\} : \sum_{i \in S} d_i \le b \right\}$$

The new model has a binary variable for each of these subsets $S \in \mathcal{S}'$

$$x_S := \begin{cases} 1, & \text{if in a solution there is bin containing the items in S} \\ 0, & \text{otherwise} \end{cases}$$

The resulting ILP model is a Set Partitioning one

$$\begin{array}{rcl} \min \sum_{S \in \mathcal{S}'} x_S & & \\ \sum_{S \in \mathcal{S}' : i \in S} x_S & = & 1, \quad i = 1, \dots, m \\ x_S & \in & \{0, 1\}, \quad S \in \mathcal{S}' \end{array} \tag{26}$$

Note that the number of variables is bounded by $O(2^m)$, i.e., huge for practical values of m

Example 2 Let us consider the case $m = 5$, $b = 10$, $d = (7, 5, 4, 4, 2)$, for which

$$\mathcal{S}' = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{4, 5\}, \{3, 4, 5\}\}$$

The resulting Set Partitioning model reads as

$$
\begin{array}{rcl}
\min \sum_{S \in \mathcal{S}'} x_S & & \\
x_{\{1\}} + x_{\{1,5\}} & = & 1 \\
x_{\{2\}} + x_{\{2,3\}} + x_{\{2,4\}} + x_{\{2,5\}} & = & 1 \\
x_{\{3\}} + x_{\{2,3\}} + x_{\{3,4\}} + x_{\{3,5\}} + x_{\{3,4,5\}} & = & 1 \\
x_{\{4\}} + x_{\{2,4\}} + x_{\{3,4\}} + x_{\{4,5\}} + x_{\{3,4,5\}} & = & 1 \\
x_{\{5\}} + x_{\{1,5\}} + x_{\{2,5\}} + x_{\{3,5\}} + x_{\{4,5\}} + x_{\{3,4,5\}} & = & 1 \\
x_S & \in & \{0,1\}, \quad S \in \mathcal{S}'
\end{array}
$$

□

It is possible to significantly reduce the number of variables of the above model by restricting the subset of items to *maximal* subsets, where a subset of items is maximal if and only if no other item can be added to the bin without exceeding the capacity. Then, collection $\mathcal{S}'$ is replaced by the following collection $\mathcal{S}$

$$
\mathcal{S} := \left\{ S \subseteq \{1, \ldots, m\} : \sum_{i \in S} d_i \leq b, \sum_{i \in S \cup \{j\}} d_i > b \ \forall j \notin S \right\}
$$

Although in general $|\mathcal{S}| << |\mathcal{S}'|$, the new model having a variable x_S for each $S \in \mathcal{S}$ is still of exponential size. However, it is easy to observe that model (26) is not valid anymore if $\mathcal{S}'$ is replaced by $\mathcal{S}$

On the other hand, a valid Set Covering-type model is obtained by replacing equations by inequalities as follows

$$
\begin{array}{rcl}
\min \sum_{S \in \mathcal{S}} x_S & & \\
\sum_{S \in \mathcal{S} : i \in S} x_S & \geq & 1, \quad i = 1, \ldots, m \\
x_S & \in & \{0,1\}, \quad S \in \mathcal{S}
\end{array}
\tag{27}
$$

Models (26) and (27) are clearly both valid and actually equivalent as proved by the following two propositions

Proposition 1 *Any solution of model* (26) *corresponds to a solution of model* (27) *with the same value*

Proof Given a solution x^* of model (26), a solution $\overline{x}$ of model (27) with the same value is obtained by considering each variable $x^*_S = 1$, determining a maximal subset $\overline{S} \in \mathcal{S}$ such that $S \subseteq \overline{S}$ and setting $\overline{x}_{\overline{S}} = 1$ □

To prove the reverse statement of Proposition 1 we need a slightly stricter condition, i.e., the minimality of the solutions of model (27): we say that a solution of model (27) is *minimal* if no bin is entirely composed by items that are packed in other bins of the solution as well. In other words, the value of a minimal solution cannot be reduced removing bins without repacking the remaining items. Then,

Proposition 2 *Any* minimal *solution of model* (27) *corresponds to a solution of model* (26) *with the same value*

Proof Given a *solution* $\overline{x}$ of model (27), a solution x^* of model (26) with the same value is obtained by Algorithm 2

Algorithm 2 Converting a minimal solution of model (27) into a solution of model (26)

$I := \emptyset$;
$\bar{S} := \{S \in \mathcal{S} : \overline{x}_S = 1\}$;
while $\bar{S} \neq \emptyset$ **do**
  $x^*_{S^*} = 1$ for $S^* := S \setminus I$;
  $I := I \cup S^*$
end while
return x^*

□

The results above hold also for the solutions of the LP relaxations of models (26) and (27) (essentially the same proofs)

On the practical side, model (27) is *better* than model (26) because

- the number of variables is smaller (although still exponential),
- the value of the LP relaxations coincide, and
- LPs with inequalities are generally easier to solve than LPs with equations

Example 3 Let us consider again the case $m = 5$, $b = 10$, $d = (7, 5, 4, 4, 2)$

$$\mathcal{S} = \{\{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4, 5\}\}$$

The resulting Set Covering model reads as

$$
\begin{array}{rcl}
\min \sum_{S\in\mathcal{S}} x_S & & \\
x_{\{1,5\}} & \geq & 1 \\
x_{\{2,3\}} + x_{\{2,4\}} + x_{\{2,5\}} & \geq & 1 \\
x_{\{2,3\}} + x_{\{3,4,5\}} & \geq & 1 \\
x_{\{2,4\}} + x_{\{3,4,5\}} & \geq & 1 \\
x_{\{1,5\}} + x_{\{2,5\}} + x_{\{3,4,5\}} & \geq & 1 \\
x_S & \in & \{0,1\}, \quad S \in \mathcal{S}
\end{array}
$$

Given the solution $x^*_{\{1,5\}} = x^*_{\{2,3\}} = x^*_{\{4\}} = 1$ of model (26), the corresponding solution of model (27) is $\overline{x}_{\{1,5\}} = \overline{x}_{\{2,3\}} = \overline{x}_{\{3,4,5\}} = 1$ $\square$

In summary, for BPP the "natural" model, although much simpler, presents several serious drawbacks

In case one needs to solve the BPP by branch and bound, and the "natural" model *out-of-the-shelf* is not enough, then one needs to use model (27), which, because of its size, has to be managed with care (see later on in the course)

1.5 Fixed Charge

The so-called *Fixed Charge* Problem arises in production planning when one has to select the mix of n products that needs to be realized so as to satisfy a demand and some other production constraints that we will generically indicate as $Ax \geq b$

Each product j is characterized by a fixed cost f_j to be payed (only once) if any quantity of product j is produced, and a cost c_j linearly depending on the quantity produced

Similarly to the UFLP, the natural nonnegative production variables

$$x_j := \text{quantity of product } j \text{ realized}$$

are not enough to model the above logical implication and the binary variables

$$y_j := \begin{cases} 1, & \text{if } x_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

have to be introduced

With the variables above the contribution to the objective function of product j is very simple to state as $f_j y_j + c_j x_j$

However, the logical implication $x_j > 0 \Rightarrow y_j = 1$ is still not obvious to express in linear terms because the x and y variables are not "comparable", i.e., any production value x_j is possible if $y_j = 1$

An elegant modeling trick that is enough to overcome this issue is writing the logical implication as

$$x_j \leq M y_j$$

where $M > 0$ is a *sufficiently*-large positive constant that deactivates the constraint in case $y_j = 1$ by imposing a loose upper bound on the production variable x_j

In this way, the overall model of the Fixed Charge Problem reads as

$$\min \sum_{j=1}^{n} f_j y_j + c_j x_j \tag{28}$$

$$x_j \leq M y_j, \quad j = 1, \ldots, n \tag{29}$$

$$Ax \geq b \tag{30}$$

$$x_j \geq 0, \quad y_j \in \{0, 1\}, \quad j = 1, \ldots, n \tag{31}$$

Constraints (29) are generally referred to as *bigM* constraints, and are largely used in Mixed-Integer Linear Programming to express logical implications

However, bigM constraints need to be managed with extreme care

Example 4 Consider the case in which the production of 100 units of product j needs to be realized, and that the value of M has been set to a very large value, say $M = 1,000,000$, to stay on the safe side

In order to satisfy the j-th constraint (29) in the LP relaxation of model (28)–(31) a value $y_j = 0.0001$ suffices □

Example 4 shows that in case the value of M has been selected only to be safe on the ILP side, i.e., to deactivate the constraints, that can result in very weak LP relaxations, thus significantly affecting the chances of solving the problem by branch and bound

Another serious risk of using bigM's is associated with precision of floating-point computation, especially in applications where the x variables can take very high values. The consequence is that M values must be very high as well, thus potentially resulting in very tiny values of the y variables in the LP relaxations. Thus, a MIP solver can erroneously conclude that a very tiny value of a y_j variable, say $y_j < 10^{-6}$, is actually integral because it is smaller than the integrality tolerance

Finally, note that constraints (6) for the UCFP are essentially bigM-type constraints. Indeed, the value m is, in general, a loose upper bound on the number of clients that can be simultaneously served by a facility, and it is used to deactivate the constraint when the associated facility has been opened. This is also the reason of the weakness of the UFLP model using constraints (6)

1.6 Stable Set and Clique

Let us consider the non-oriented graph $G = (V, E)$ with *weight* p_j for each vertex $j \in V$, and indicate with $n := |V|$ the number of vertices and with $m := |E|$ the numbers of edges

A *Stable Set* (or *Indpendent Set*) of G is a subset of vertices $S \subseteq V$ such that $E(S) = \emptyset$, i.e., no edge in E connects two vertices in S directly

The (maximum-weight) *Stable Set* (or *Independent Set*, or *Vertex/Node Packing*) problem calls for determining the Stable Set of G of *maximum* weight

A simple ILP model with m linear constraints for the Sable Set problem is obtained by using the following (natural) binary variables

$$x_j := \begin{cases} 1, & \text{if vertex } j \text{ belongs to the stable set} \\ 0, & \text{otherwise} \end{cases}$$

and reads as

$$\max \sum_{j \in V} p_j x_j \tag{32}$$

$$x_i + x_j \leq 1, \quad (i, j) \in E \tag{33}$$

$$x_j \in \{0, 1\}, \quad j \in V \tag{34}$$

The "weakness" of the associated LP relaxation is obvious by considering the trivial special case of a complete graph G and, for example, $p_j = 1$ for each $j \in V$. For such a family of instances

- the optimal solution of the ILP is equal to 1 (only one vertex in the stable set), while
- the optimal solution of the LP relaxation is $x_j = 1/2$ per $j \in V$, and has value $n/2$

A much "stronger" model is obtained by exploiting the notion of *Clique* of G, which corresponds to a subset of vertices $K \subseteq V$ such that $E(K) = \{(i, j) : i, j \in K\}$, i.e., all pairs of vertices in S are *directly* connected by an edge in E

A clique K is said to be *maximal* if and only if it does not exist a clique K' such that $K \subset K'$, i.e., it does not exist another vertex in $V \setminus K$ directly connected to every vertex of K by an edge in E

By letting $\mathcal{K}$ indicate the collection of all maximal cliques of G, and by observing that each stable set can contain at most one vertex in each clique, a strong model (at the price of an exponential number $|\mathcal{K}| = O(2^n)$ of linear constraints) is obtained by replacing constraints (33) by

$$\sum_{j \in K} x_j \leq 1, \quad K \in \mathcal{K} \tag{35}$$

Of course, like in the case of model (27) for BPP involving an exponential number of variables, also this model has to be managed with care

A sort of "intermediate" model between the two above, with not more than m constraints like the "weak" model, but all of type (35), like the "strong" model, is obtained, starting from the "weak" model, by

- replacing each constraint $x_i + x_j \leq 1$ with a constraint $\sum_{j \in K} x_j \leq 1$ for *some* clique $K \in \mathcal{K}$ such that $i, j \in K$ (where it is easy to see that a similar clique is easy to obtain), and
- removing possibly duplicated constraints

The resulting ILP model is defined by (32), (34) and

$$\sum_{j \in K} x_j \leq 1, \quad \text{for } \textit{each } (i,j) \in E \text{ and for } \textit{some } K \in \mathcal{K} \text{ such that } i, j \in K \tag{36}$$

Example 5 For graph $G = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 5), (2, 6), (5, 6)\}$ constraints (33) are

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ x_1 + x_3 &\leq 1 \\ x_1 + x_4 &\leq 1 \\ x_1 + x_5 &\leq 1 \\ x_1 + x_6 &\leq 1 \\ x_2 + x_5 &\leq 1 \\ x_2 + x_6 &\leq 1 \\ x_5 + x_6 &\leq 1 \end{aligned}$$

while $\mathcal{K} = \{\{1, 2, 5, 6\}, \{1, 3\}, \{1, 4\}\}$, thus constraints (35) are

$$\begin{aligned} x_1 + x_2 + x_5 + x_6 &\leq 1 \\ x_1 + x_3 &\leq 1 \\ x_1 + x_4 &\leq 1 \end{aligned}$$

Finally, for such a simple example constraints (36) coincide with constraints (35) $\square$

1.6.1 Stable Set and Set Packing

It is easy to show that the Stable Set problem and the Set Packing problem as defined by (12) are actually the same problem

First, all ILP models introduced in the previous section for the Stable Set problem are of Set Packing type

Vice versa

Proposition 3 *The Set Packing problem* (12) *associated with the constraint matrix* $A \in \{0,1\}^{m \times n}$ *and the cost vector* c *is equivalent to the Stable Set problem associated with the undirected graph* $G(A) = (V, E)$ *with vertex set* $V := \{1, \dots, n\}$*, weights* $p_j := c_j$ *for* $j \in V$*, and edge set*

$$E := \{(i,j) : a_{hi} = a_{hj} = 1 \text{ for some } h \in \{1, \dots, m\}\}$$

Proof Two variables x_i and x_j can take both value 1 in a Set Packing solution if and only if it does not exist a constraint h such that $a_{hi} = a_{hj} = 1$, i.e., if and only if vertices $i, j \in G(A)$ are not (directly) connected by an edge

That implies that any solution of Set Packing corresponds to a Stable Set in $G(A)$ and vice versa $\square$

Visualizing a Set Packing problem on a graph is of fundamental importance for verifying that the corresponding inequalities are "strong", and to strengthen them otherwise

Example 6 Consider the following Set Packing problem

$$\begin{array}{rcl}
\max \sum_{j=1}^{6} c_j x_j & & \\
x_1 + x_4 + x_6 & \leq & 1 \\
x_2 + x_4 + x_5 & \leq & 1 \\
x_3 + x_4 & \leq & 1 \\
x_2 + x_3 + x_5 & \leq & 1 \\
x_j & \in & \{0,1\}, \quad j = 1, \dots, 6
\end{array}$$

It is possible to define an associated undirected graph $G(A) = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1,4), (1,6), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5), (4,6)\}$

The first constraint of the ILP model above corresponds to the maximal clique $\{1, 4, 6\}$ of $G(A)$. In other words, that constraint is already "strong". However, the remaining three constraints can be replaced by the following inequality that corresponds to the maximal clique $\{2, 3, 4, 5\}$ of $G(A)$, with the result of strengthening the model

$$x_2 + x_3 + x_4 + x_5 \leq 1$$

$\square$

1.6.2 Clique Inequalities and (M)ILP

Generally speaking, Algorithm 3 is fundamental for strengthening a set of inequalities that appear in a generic (M)ILP model in the so-called *clique* form

$$\sum_{j \in S} x_j \leq 1 \tag{37}$$

where x_j is a binary variable for all $j \in S$

Algorithm 3 Strengthening Set Packing constraints in a general MILP

define the undirected graph $G = (V, E)$ according to Proposition 3;
let I be the set of linear inequalities (37) to be strengthened;
for all $i \in I$ **do**
    let S_i be the set of binary variables involved in the clique inequality i;
    if S_i is *not* a maximal clique of G **then**
        find a maximal clique S_i' such that $S_i \subset S_i'$;
        replace inequality i with $\sum_{j \in S_i'} x_j \leq 1$
    end if
end for

1.7 Vertex Coloring

Given an undirected graph $G = (V, E)$ with $n := |V|$ vertices and $m := |E|$ edges, the so-called *Vertex Coloring* Problem calls for assigning *colors* to the vertices of G such that

- vertices directly connected by an edge in E receive different colors, and
- the number of used colors is a minimum

Seemingly to the natural model for the BPP, and observing that n colors are always enough (and necessary if and only if the graph is complete), one can introduce binary variables

$$y_j := \begin{cases} 1, & \text{if color } j \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij} := \begin{cases} 1, & \text{if vertex } i \text{ is colored by color } j \\ 0, & \text{otherwise} \end{cases}$$

Then, the simplest model for the Vertex Coloring Problem reads as

$$\min \sum_{j=1}^{n} y_j \tag{38}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i \in V \tag{39}$$

$$x_{ij} + x_{hj} \leq y_j, \quad (i,h) \in E, \ j = 1, \ldots, n \tag{40}$$

$$y_j, x_{ij} \in \{0,1\}, \quad i \in V, \ j = 1, \ldots, n \tag{41}$$

The above model combines the drawbacks of the natural model of BPP, and those of the "weak" model of the Stable Set Problem, as shown, for example, by the special case where G is complete where

- the optimal solution value of the ILP is n (a different color for each vertex), while
- the optimal solution of the LP relaxation is $y_1 = y_2 = 1$; $x_{i1} = x_{i2} = 1/2, \forall i \in V$, and has value 2

As for the Stable Set, the model can be strengthened by replacing constraints (40) by constraints

$$\sum_{i \in K} x_{ij} \leq y_j, \quad K \in \mathcal{K}, \ j = 1, \ldots, n \tag{42}$$

where again $\mathcal{K}$ denotes the collection of all maximal cliques of G

However, the drawbacks associated with the natural model of BPP remain, together with the fact that there are $O(n2^n)$ inequalities

For example, the optimal solution value of the LP relaxation of the strengthened model is equal to the size of the clique of G with *largest cardinality*, which is a trivial lower bound on the minimum number of required colors

Seemingly to BPP, an alternative ILP model is obtained by observing that the set of vertices that receive the same color in any solution of the Vertex Coloring Problem corresponds to a Stable Set of G

By considering the collection $\mathcal{S}$ of all *maximal* Stable Sets of G, and by introducing a binary variable for each of them

$$x_S := \begin{cases} 1, & \text{if all and only the vertices in } S \text{ receive the same color in a solution} \\ 0, & \text{otherwise} \end{cases}$$

a Set Covering-type ILP model, similar to model (27) for BPP, is the following

$$\begin{aligned} \min \sum_{S \in \mathcal{S}} x_S & \\ \sum_{S \in \mathcal{S} : i \in S} x_S & \geq 1, \quad i \in V \\ x_S & \in \{0,1\}, \quad S \in \mathcal{S} \end{aligned} \tag{43}$$

It is easy to devise the corresponding Set Partitioning-type model, with a variable for each (not necessarily maximal) Stable Set of G by applying the same reasoning done for BPP

Example 7 Given the graph $G = (V, E)$ with $V = \{1,2,3,4,5,6\}$ and $E = \{(1,2),(1,3),$ $(1,4),(1,5),(1,6),(2,5),(2,6),(5,6)\}$, then $\mathcal{S} = \{\{1\},\{2,3,4\},\{3,4,5\},\{3,4,6\}\}$, and the associated ILP model (43) is

$$\begin{aligned} \min \textstyle\sum_{S \in \mathcal{S}} x_S & \\ x_{\{1\}} & \geq 1 \\ x_{\{2,3,4\}} & \geq 1 \\ x_{\{2,3,4\}} + x_{\{3,4,5\}} + x_{\{3,4,6\}} & \geq 1 \\ x_{\{2,3,4\}} + x_{\{3,4,5\}} + x_{\{3,4,6\}} & \geq 1 \\ x_{\{3,4,5\}} & \geq 1 \\ x_{\{3,4,6\}} & \geq 1 \\ x_S & \in \{0,1\}, \quad S \in \mathcal{S} \end{aligned}$$

□

Finally, observe that the dual of the continuous relaxation of (43) is

$$
\begin{aligned}
\max \sum_{i \in V} y_i & & & \\
\sum_{i \in S} y_i & \leq & 1, & \quad S \in \mathcal{S} \\
y_i & \geq & 0, & \quad i \in V
\end{aligned}
\tag{44}
$$

and corresponds to the continuous relaxation of the model for the problem of determining the maximum-weight *Clique* in G (with all weights equal to 1, i.e., the clique of maximum cardinality), equivalent to the model (32), (35), (34) for the problem of determining the maximum-weight Stable Set

More precisely, the Clique problem in a graph $G = (V, E)$ coincides with the Stable Set problem on the graph $\overline{G} = (V, \overline{E})$, called *complement graph* of G, having the same set of vertices V of G and all and only the edges that G misses, namely

$$\overline{E} := \{(i, j) : i, j \in V, i \neq j, (i, j) \notin E\}$$

1.8 Traveling Salesman

The so-called *Traveling Salesman* Problem (TSP) is the most celebrated problem in combinatorial optimization. It is usually defined on an undirected graph, but because the version on a directed graph, called *Asymmetric TSP* (ATSP), admits an ILP model that is more general we will consider the ATSP first

Given a directed graph $G = (V, A)$, complete and with cost $c_a = c_{(i,j)}$ for each arc $a = (i, j) \in A$, the ATSP calls for determining a *tour* of G

- visiting each vertex $i \in V$ *exactly once*, and
- at a minimum cost, where the cost of a tour is the sum of the costs of its arcs

Observe that the requirement of visiting each vertex exactly once, i.e., once but *not more*, which seems to be unrealistic for routing problems (in case it is convenient to go through a vertex more than once it should be possible) is actually *redundant* in case the costs satisfy the *triangular inequality*

$$c_{(i,j)} + c_{(j,k)} \geq c_{(i,k)}, \quad i, j, k \in V, i \neq j, i \neq k, j \neq k \tag{45}$$

as it is often the case in practice. Indeed, in many applications the cost matrix c is actually obtained by computing the shortest path between each pair of vertices

For the ATSP (as well as for the TSP) it exists a unique ILP model that is successfully used in practice and has the following binary variables

$$x_a := \begin{cases} 1, & \text{if arc } a \text{ belongs to the tour} \\ 0, & \text{otherwise} \end{cases}$$

It is easy to observe that a tour visiting all vertices exactly once has precisely one incoming and one outgoing arc for each vertex of G. Thus, one could think that the ILP model

$$\min \sum_{a \in A} c_a x_a \tag{46}$$

$$\sum_{a \in \delta^-(i)} x_a = 1, \quad i \in V \tag{47}$$

$$\sum_{a \in \delta^+(i)} x_a = 1, \quad i \in V \tag{48}$$

$$x_a \in \{0, 1\}, \quad a \in A \tag{49}$$

is enough to express the ATSP

Example 8 For a graph of 6 vertices, constraints (47) and (48), the so-called *degree* constraints, take the form

$$\begin{aligned}
x_{(2,1)} + x_{(3,1)} + x_{(4,1)} + x_{(5,1)} + x_{(6,1)} &= 1 \\
\dots & \\
x_{(1,6)} + x_{(2,6)} + x_{(3,6)} + x_{(4,6)} + x_{(5,6)} &= 1 \\
x_{(1,2)} + x_{(1,3)} + x_{(1,4)} + x_{(1,5)} + x_{(1,6)} &= 1 \\
\dots & \\
x_{(6,1)} + x_{(6,2)} + x_{(6,3)} + x_{(6,4)} + x_{(6,5)} &= 1
\end{aligned}$$

□

However, it is easy to see that a solution of model (46)–(49) might have *more than one tour*

In other words, the model above is *not valid* for the ATSP, while it precisely models another very well-known problem in combinatorial optimization, the so-called *Assignment* Problem (AP). The quite peculiar characteristic of the AP is that the LP relaxation of model (46)–(49) defines the convex hull of its integer solutions (thus, in turn, showing that AP is polynomially solvable)

In order to obtain a valid ILP model for the ATSP we need to add constraints forbidding the existence of *subtours*, i.e., tours visiting only a subset of the vertices

Let $\mathcal{C}$ be the collection of all subtours of G, then a set of constraints that added to model (46)–(49) define a valid ILP model for the ATSP is

$$\sum_{a \in C} x_a \leq |C| - 1, \quad C \in \mathcal{C} \tag{50}$$

It is easy to observe that the number of subtours of k arcs, i.e., visiting k vertices, of a graph with n vertices is equal to $\binom{n}{k}(k-1)!$, because there are $\binom{n}{k}$ ways of selecting the k visited vertices and, for each of those choices, $(k-1)!$ ways of defining a subtour visiting them

Example 9 For a graph of 6 vertices, constraints (50) have the form

$$
\begin{array}{rcl}
x_{(1,2)} + x_{(2,1)} & \leq & 1 \\
x_{(1,3)} + x_{(3,1)} & \leq & 1 \\
\ldots & & \\
x_{(1,2)} + x_{(2,3)} + x_{(3,1)} & \leq & 2 \\
x_{(1,3)} + x_{(3,2)} + x_{(2,1)} & \leq & 2 \\
\ldots & & \\
x_{(1,2)} + x_{(2,3)} + x_{(3,4)} + x_{(4,1)} & \leq & 3 \\
x_{(1,2)} + x_{(2,4)} + x_{(4,3)} + x_{(3,1)} & \leq & 3 \\
x_{(1,3)} + x_{(3,2)} + x_{(2,4)} + x_{(4,1)} & \leq & 3 \\
x_{(1,3)} + x_{(3,4)} + x_{(4,2)} + x_{(2,1)} & \leq & 3 \\
x_{(1,4)} + x_{(4,2)} + x_{(2,3)} + x_{(3,1)} & \leq & 3 \\
x_{(1,4)} + x_{(4,3)} + x_{(3,2)} + x_{(2,1)} & \leq & 3 \\
\ldots & &
\end{array}
$$

□

It exists a much better version of constraints (50), which is the one used in practice because the set contains less constraints that are actually stronger

Given a subset of vertices $S \subseteq V$ and a subtour visiting the vertices in S, constraints (50) require that at most $|S| - 1$ arcs *of the* subtour can be selected in a solution. However, is is easy to see that a stronger condition holds: at most $|S| - 1$ arcs *between pairs of vertices in* S can be selected in a solution (otherwise, subtours would appear)

The above observation leads to stronger constraints forbidding subtours, the so-called *Subtour Elimination* Constraints

$$
\sum_{a \in A(S)} x_a \quad \leq \quad |S| - 1, \quad S \subseteq V, 2 \leq |S| \leq |V| - 2 \tag{51}
$$

Observe that it is not necessary to introduce constraints (51) if $|S| = 1$ and $|S| = |V| - 1$ because no solution of model (46)–(49) can contain a subtour that visits only one vertex

For each subset $S \subseteq V$ such that $2 \leq |S| \leq |V| - 2$, the *unique* constraint (51) associated with S *dominates* the $(|S| - 1)!$ constraints (50) associated with S

Then, the (final) ILP model for the ATSP is given by the degree constraints (46)–(49) together with the $2^n - 2(n + 1)$ constraints (51)

Example 10 For a graph of 6 vertices, the constraints (51) that replace constraints (50) of Example 9 have the form

$$
\begin{array}{rcl}
x_{(1,2)} + x_{(2,1)} & \leq & 1 \\
x_{(1,3)} + x_{(3,1)} & \leq & 1 \\
\dots & & \\
x_{(1,2)} + x_{(1,3)} + x_{(2,1)} + x_{(2,3)} + x_{(3,1)} + x_{(3,2)} & \leq & 2 \\
\dots & & \\
x_{(1,2)} + x_{(1,3)} + x_{(1,4)} + x_{(2,1)} + x_{(2,3)} + x_{(2,4)} + x_{(3,1)} + x_{(3,2)} + x_{(3,4)} + x_{(4,1)} + x_{(4,2)} + x_{(4,3)} & \leq & 3 \\
\dots & &
\end{array}
$$

□

It exists an alternative and equivalent way of expressing constraints (51) as

$$\sum_{a \in \delta^+(S)} x_a \quad \geq \quad 1, \quad S \subseteq V, 2 \leq |S| \leq |V| - 2 \tag{52}$$

The equivalence is stated in the following Proposition

Proposition 4 *For a vector $x = (x_a)$ satisfying* (47) *and* (48)*, x satisfies* (51) *if and only if it satisfies* (52) *as well*

Proof Consider x that satisfies (47) and (48) and a generic subset $S \subseteq V$ such that $2 \leq |S| \leq |V| - 2$
First, identity

$$\sum_{i \in S} \sum_{a \in \delta^+(i)} x_a = \sum_{a \in A(S)} x_a + \sum_{a \in \delta^+(S)} x_a$$

holds. Moreover, because x satisfies (48), then

$$\sum_{i \in S} \sum_{a \in \delta^+(i)} x_a = |S|$$

By combining the two equations above, then

$$\sum_{a \in A(S)} x_a + \sum_{a \in \delta^+(S)} x_a = |S|$$

that, in turn, implies

$$\sum_{a \in A(S)} x_a \leq |S| - 1 \Leftrightarrow \sum_{a \in \delta^+(S)} x_a \geq 1$$

that is, x satisfies constraint (51) for S if and only if satisfies (52) for S as well □

Example 11 For a graph of 6 vertices, constraints (52) corresponding to constraints (51) in Example 10 have the form

$$\begin{array}{rcl}
x_{(1,3)} + x_{(1,4)} + x_{(1,5)} + x_{(1,6)} + x_{(2,3)} + x_{(2,4)} + x_{(2,5)} + x_{(2,6)} & \geq & 1 \\
x_{(1,2)} + x_{(1,4)} + x_{(1,5)} + x_{(1,6)} + x_{(3,2)} + x_{(3,4)} + x_{(3,5)} + x_{(3,6)} & \geq & 1 \\
\dots & & \\
x_{(1,4)} + x_{(1,5)} + x_{(1,6)} + x_{(2,4)} + x_{(2,5)} + x_{(2,6)} + x_{(3,4)} + x_{(3,5)} + x_{(3,6)} & \geq & 1 \\
\dots & & \\
x_{(1,5)} + x_{(1,6)} + x_{(2,5)} + x_{(2,6)} + x_{(3,5)} + x_{(3,6)} + x_{(4,5)} + x_{(4,6)} & \geq & 1 \\
\dots & &
\end{array}$$

□

Observe that the equivalence between (51) and (52) is true because of the degree constraints, while there are TSP variants in which one is looking for a circuit not necessarily visiting all vertices, where only either (51) or (52) are satisfied

The TSP is the ATSP variant defined on an undirected graph, i.e., on $G = (V, E)$, complete and with cost $c_e = c_{(i,j)}$ for each edge $e = (i, j) \in E$

Clearly, the TSP is the special case of the ATSP where $c_{(i,j)} = c_{(j,i)}$ for $i, j \in V, i \neq j$, thus the ATSP model above is valid for the TSP as well

However, the TSP is generally solved through an ILP model where variables are associated with edges of G (instead of arcs)

$$x_e := \begin{cases} 1, & \text{if edge } e \text{ belongs to the tour} \\ 0, & \text{otherwise} \end{cases}$$

and reads as follows

$$\min \sum_{e \in E} c_e x_e \tag{53}$$

$$\sum_{e \in \delta(i)} x_e \; = \; 2, \quad i \in V \tag{54}$$

$$\sum_{e \in E(S)} x_e \; \leq \; |S| - 1, \quad S \subseteq V, 2 \leq |S| \leq |V| - 2 \tag{55}$$

$$x_e \; \in \; \{0, 1\}, \quad e \in E \tag{56}$$

Note that, for $|S| = 2$, the constraint (55) associated with $S = \{i, j\}$ is $x_{(i,j)} \leq 1$. These are redundant constraints for the integer case but necessary for the LP relaxation

The alternative to constraints (55), equivalent as for the ATSP, is writing constraints

$$\sum_{e \in \delta(S)} x_e \; \geq \; 2, \quad S \subseteq V, 2 \leq |S| \leq |V| - 2 \tag{57}$$

1.9 Summary of Problems and ILP Models

The following table summarizes the problems introduced and their discussed ILP models

Problem	ILP model	(#variables, #constraints)	The associated LP relaxation is
UFLP	(1), (2), (6), (4)	$(n+mn,\ m+n)$	"weak"
	(1), (2), (5), (4)	$(n+mn,\ m+mn)$	"strong"
Set Covering	(10)	$(n,\ m)$	"strong"
Set Partitioning	(11)	$(n,\ m)$	"strong"
Set Packing	(12)	$(n,\ m)$	"intermediate" if all constraints are associated with maximal cliques of $G(A)$ (see Stable Set)
CFLP	(13)-(17)	$(n+mn,\ m+mn+n)$	generally "strong", even if tends to become "weak" if facilities are equal (see Bin Packing)
Bin Packing	(18)-(22)	$(n+mn,\ m+mn+n)$	"weak"
	(27)	$(O(2^n),\ n)$	"strong"
Fixed Charge	(29)	$(2n,\ n)$ plus $Ax \geq b$	"weak" depending on M values
Stable Set	(32)-(34)	$(n,\ m)$	"weak"
	(32), (35), (34)	$(n,\ O(2^n))$	"strong"
	(32), (36), (34)	$(n,\ O(m))$	"intermediate"
Vertex Coloring	(38)-(41)	$(n+n^2,\ n+nm)$	"weak"
	(38), (39), (42), (41)	$(n+n^2,\ n+O(n2^n))$	"intermediate"
	(43)	$(O(2^n),\ n)$	"strong"
ATSP	(46)-(49), (51)	$(n(n-1),\ 2n+O(2^n))$	"strong"
TSP	(53)-(56)	$(n(n-1)/2,\ n+O(2^n))$	"strong"

2 Solving Models of Exponential Size

In this section we consider the models that are "too large" to be solved by a black-box MILP solver, in particular because their LP relaxation has exponentially-many constraints or variables

2.1 Models with (too) Many Constraints and Separation

First, we consider models whose LP relaxation is of exponential size in the number of constraints as, for example, the continuous relaxations of ATSP (46)-(49), (51) and Stable Set (32), (35), (34)

In general, we consider the generic LP

$$\begin{array}{rcl} \min c^T x & & \\ Ax & \geq & b \\ x & \geq & 0 \end{array} \tag{58}$$

where the number of inequalities $Ax \geq b$ is huge

The algorithm for solving the problem *without* feeding the LP solver with all constraints is

1. Initialize $\tilde{A}$, $\tilde{b}$ with a ("small") subset of rows in A, b

2. Solve the associated "reduced" LP relaxation

$$\begin{array}{rcl} \min c^T x & & \\ \tilde{A}x & \geq & \tilde{b} \\ x & \geq & 0 \end{array} \tag{59}$$

   and get the solution x^*

3. If x^* satisfies all constraints $Ax \geq b$, then *STOP* (x^* is the optimal solution of (58)). Otherwise, add to $\tilde{A}$, $\tilde{b}$ the rows corresponding to *some* violated constraints, and go to 2

It is easy to see that the above scheme is rather simple and direct: it adds the constraints "on-the-fly", i.e., when they are needed because they are violated

Despite its simplicity the algorithm works very well in practice in the sense that the final number of constraints in $\tilde{A}x \geq \tilde{b}$ is generally much smaller than the total number. This can be proven theoretically, which is, however, outside of the scope of these notes

The above algorithm, as presented, has two degrees of freedom, one associated with the initial subset of constraints to be considered and the other in the choice of the (violated) constraints to be added at each iteration

In practice, these choices are rarely *critical*, in the sense that the algorithm generally converges quickly independently of the initial subset of constraints (provided the initial LP is *bounded*) and even adding only one violated constraint at each step

Instead, the crucial part of the algorithm is how to effectively find the violated constraints at step 3. This step is generally called *Separation*

Of course, simply enumerating the constraints exhaustively is unpractical precisely like considering them all together. Alternatively, a specific (optimization) problem has to be defined and solved, often by ILP, as illustrated in the ATSP and Stable Set examples below

(It might look weird and inefficient to solve ILPs only for separating violated constraints used in turn to solve an LP, which is, most of the time, the relaxation of another ILP. Besides some fundamental theoretical results that corroborates the use of this approach, it also works in practice sometimes even if the separation ILPs are solved by a black-box solver, i.e., without exploiting specific knowledge on the problem at hand)

It is easy to observe that during the "row generation" algorithm above the optimal value $c^T x^*$ of the reduced LP is always smaller than (or equal to) that of the full LP (58). Thus, it is a valid lower bound on the optimal value of the ILP associated with (58), if any

This observation implies that even if the "row generation" algorithm is prematurely interrupted (because of a time limit or because the separation problem is solved heuristically and no more violated constraints are found) a valid lower bound is available

2.1.1 ATSP

The first example we consider is that of solving the LP relaxation of the ATSP model, namely system (46)-(49), (51)

It is natural to use as initial subset of constraints for step 1 the so-called "degree" constraints (47) and (48)

Thus, the separation problem of step 3 calls for finding subtour elimination constraints (51) violated by a solution x^* obtained at step 2, and can be formulated as

Given $x^* = (x^*_a)$, find, if any, a subset $S^* \subseteq V$ such that

- $\sum_{a \in A(S^*)} x^*_a > |S^*| - 1 \Leftrightarrow |S^*| - \sum_{a \in A(S^*)} x^*_a < 1$
- $2 \leq |S^*| \leq |V| - 2$

Instead of explicitly considering all possible subsets S^* (unpractical for values of $|V|$ not too small), the separation problem can be formulated (as an ILP) by introducing binary variables

$$y_i := \begin{cases} 1, & \text{if vertex } i \text{ belongs to } S^* \\ 0, & \text{otherwise} \end{cases}$$

and

$$z_a := \begin{cases} 1, & \text{if arc } a \text{ belongs to } A(S^*) \\ 0, & \text{otherwise} \end{cases}$$

and solved by finding, if any, an integer solution of system

$$\begin{aligned}
\sum_{i \in V} y_i - \sum_{a \in A} x_a^* z_a \quad &< \quad 1 && (60)\\
z_{(i,j)} = 1 \quad \Leftrightarrow \quad & y_i = y_j = 1, \quad (i,j) \in A && (61)\\
\sum_{i \in V} y_i \quad &\geq \quad 2 && (62)\\
\sum_{i \in V} y_i \quad &\leq \quad |V| - 2 && (63)\\
y_i, z_a \quad &\in \quad \{0,1\}, \quad i \in V,\ a \in A && (64)
\end{aligned}$$

(Actually, it would be easy to show that constraints (62) and (63) are not necessary)

Note that in the separation problem $x_a^*, \forall a \in A$ are *given values* and *not* variables

The second, logical, constraint (61) can be expressed in linear terms by the following three inequalities

$$\begin{aligned}
z_{(i,j)} \quad &\leq \quad y_i, \quad (i,j) \in A && (65)\\
z_{(i,j)} \quad &\leq \quad y_j, \quad (i,j) \in A && (66)\\
z_{(i,j)} \quad &\geq \quad y_i + y_j - 1, \quad (i,j) \in A && (67)
\end{aligned}$$

Moreover, the strict inequality (60) can be moved into the objective function

$$\min \sum_{i \in V} y_i - \sum_{a \in A} x_a^* z_a \qquad (68)$$

Then, after having solved the complete ILP (68), (62)-(64), (65)-(67), if the optimal solution $(\overline{y}, \overline{z})$ takes a value ≥ 1, this means that *all* constraints (51) are satisfied by the given x^*. Otherwise, the subset $S^* := \{i \in V : \overline{y}_i = 1\}$ defines the constraint (51) that is *most violated* by x^*, and that is added to the current set of (active) constraints in step 3

Example 12 Consider the complete directed graph with 9 vertices and cost equal to 1 for all arcs but those in $A(\{1,2,3\}) \cup A(\{4,5,6\}) \cup A(\{7,8,9\}) \cup A(\{3,6,9\})$ whose cost is equal to 0

By using in the initial set of constraints $\tilde{A}x \geq \tilde{b}$ the 18 degree equations (47) and (48), the "row generation" algorithm evolves as

1. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(2,3),(3,1),(4,5),(5,6),(6,4),(7,8),(8,9),(9,7)\}$$
and the constraint (51) associated with $S^* = \{1,2,3\}$ is violated and added to $\tilde{A}x \geq \tilde{b}$
2. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(2,1),(3,6),(6,9),(9,3),(4,5),(5,4),(7,8),(8,7)\}$$
and the constraint (51) associated with $S^* = \{1,2\}$ is violated and added to $\tilde{A}x \geq \tilde{b}$
3. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(2,3),(3,9),(9,8),(8,7),(7,1),(4,5),(5,6),(6,4)\}$$
and the constraint (51) associated with $S^* = \{4,5,6\}$ is violated and added to $\tilde{A}x \geq \tilde{b}$
4. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(2,3),(3,6),(6,9),(9,1),(4,5),(5,4),(7,8),(8,7)\}$$
and the constraint (51) associated with $S^* = \{4,5\}$ is violated and added to $\tilde{A}x \geq \tilde{b}$
5. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(2,3),(3,6),(6,5),(5,4),(4,1),(7,8),(8,9),(9,7)\}$$
and the constraint (51) associated with $S^* = \{7,8,9\}$ is violated and added to $\tilde{A}x \geq \tilde{b}$
6. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(2,3),(3,9),(9,6),(6,5),(5,4),(4,1),(7,8),(8,7)\}$$
and the constraint (51) associated with $S^* = \{7,8\}$ is violated and added to $\tilde{A}x \geq \tilde{b}$
7. the nonzero (positive) components x^* are $x^*_a = 1$ for
$$a \in \{(1,2),(4,5),(7,8)\}$$
and $x^*_a = 1/2$ for
$$a \in \{(2,3),(2,4),(3,1),(3,6),(5,6),(5,7),(6,4),(6,9),(8,1),(8,9),(9,3),(9,7)\}$$
and no constraint (51) is violated, thus x^* is also the optimal solution of the full LP

Before ending this part it is worth noting that in the special case of the separation of the subtour elimination constraints for the ATSP it is possible to formulate it as a *maximum flow* problem on an induced graph. This means in turn that the separation problem is polynomially solvable

2.1.2 Stable Set

The second example we consider is that of solving the continuous relaxation of the "strong" formulation of the Stable Set problem (32), (35), (34)

At step 1, it is natural to start with the subset of the clique constraints (35) that correspond to the "intermediate" formulation (36)

The separation problem at step 3 clearly calls for finding constraints (35) violated by x^*, and can be formulated as

Given $x^* = (x^*_a) \geq 0$, find, if any, a subset $K^* \in \mathcal{K}$ such that $\sum_{j \in K^*} x^*_j > 1$

Again, instead of considering explicitly all possible subsets $K \in \mathcal{K}$ (unpractical for values of $|V|$ not too small), the separation problem can be solved by finding the *maximum-weight Clique* in the same graph with weights $(x^*_j), \forall j \in V$. To avoid formulating this problem with exponentially-many constraints, one can use the "intermediate" ILP model, which is perfectly equivalent to the Stable Set one

Let us denote as K^* the maximum-weight clique with weights (x^*_j), if $\sum_{j \in K^*} x^*_j \leq 1$, then all constraints (35) are satisfied by x^*, which is the optimal solution of the full LP. Otherwise, any maximal clique K' such that $K^* \subseteq K'$ (obviously $K' = K^*$ if K^* is maximal) gives the constraint (35) that is *most violated* by x^*

2.2 Models with (too) Many Columns and Column Generation

We now consider LPs of exponential size in the number of variables as the continuous relaxation of formulation (27) of the Bin Packing and formulation (43) of the Vertex Coloring

Let us consider again the generic LP

$$\begin{array}{rcl} \min c^T x & & \\ Ax & \geq & b \\ x & \geq & 0 \end{array} \tag{69}$$

where now the number of variable is huge

On the contrary of the case of considering a subset of the constraints that requires to test the *feasibility* of the solution of the reduced LP, if one considers only a subset of the variables one has to test the *optimality* of the solution of the reduced LP

The optimality test is possible by exploiting the *dual problem* of LP (69)

Namely, the dual of (69) is

$$\begin{array}{rcl} \max b^T y & & \\ A^T y & \leq & c \\ y & \geq & 0 \end{array} \tag{70}$$

Precisely, given a pair of primal/dual solutions of the reduced LP having only a subset of the variables (i.e., a subset of the constraints in the dual), the pair is optimal for the full LP if and only if the dual solution is feasible for it

The algorithm for solving (69) without explicitly giving all variables to the LP solver is essentially the "dual" of the algorithm we have seen for LPs with many constraints, or, equivalently, the "row generation" algorithm applied to formulation (70). Namely,

1. Initialize $\tilde{A}$, $\tilde{c}^T$ e $\tilde{x}^T$ with a ("small") subset of columns of A, c^T, x^T
2. Solve the (reduced) LP with such a subset of variables/columns

$$\begin{array}{rcl} \min \tilde{c}^T \tilde{x} & & \\ \tilde{A}\tilde{x} & \geq & b \\ \tilde{x} & \geq & 0 \end{array} \tag{71}$$

   and get the solution $\tilde{x}^*$ and the associated dual solution y^*

3. If y^* satisfies all constraints in $A^T y \leq c$, then *STOP* ($\tilde{x}^*$ defines the optimal solution of (69), where the value of all variables that do not appear in (71) is set to 0). Otherwise, add to $\tilde{A}$, $\tilde{c}^T$, $\tilde{x}^T$ the columns corresponding to some of the violated dual constraints, and go to 2

In this case too, the choice of the initial LP is not practically "critical", provided the initial reduced LP is *feasible*, i.e., it is possible to satisfy all the constraints without using a subset of the variables. Note that considering only a subset of the variables is equivalent to set the others to 0

The separation problem on the dual at step 3 is called *Column Generation*, and it might be modeled as a specific (optimization) problem and then solved often by an ILP solver. This is illustrated in the following by two examples

Given a solution y^* at step 3 of the "column generation" algorithm, the quantity $c_j - \sum_{i=1}^m a_{ij} y_i^*$ is called *reduced cost* of variable x_j. Thus, a constraint is violated if and only if the associated reduced cost is negative

Note that during the "column generation" algorithm the optimal value $\tilde{c}^T \tilde{x}^*$ is always *greater than* (or equal to) the one of the full LP, thus we do not know if it is bigger or smaller than the optimal value of the ILP of which the full LP is the continuous relaxation, if any

Based on the above observation, it is easy to see that if the "column generation" algorithm is stopped prematurely (because of a time limit or because the separation problem of the dual is solved heuristically and no more violated constraints are found), then no lower bound is available

2.2.1 Bin Packing

The first "column generation" example we consider is the one for solving the continuous relaxation of the Bin Packing model (27). The associated dual is

$$
\begin{array}{rcll}
\max \sum_{i=1}^{m} y_i & & & \\
\sum_{i \in S} y_i & \leq & 1, & S \in \mathcal{S} \\
y_i & \geq & 0, & i = 1, \ldots, m
\end{array} \tag{72}
$$

At step 1, it is natural to construct the initial (reduced) LP by using the variables associated with a heuristic bin packing solution, after having possibly made maximal the subsets of objects that are packed in each of the bins

Then, the "column generation" problem that has to be solved at step 3 calls for finding constraints of type (72) violated by y^*, and, by explicitly defining $\mathcal{S}$, can be formulated as

Given $y^* = (y_i^*) \geq 0$, find, if any, a subset $S^* \subseteq \{1, \ldots, m\}$ such that

- $\sum_{i \in S^*} y_i^* > 1$
- $\sum_{i \in S^*} d_i \leq b$
- S^* is maximal with respect to the condition above

By initially relaxing the condition on the maximality of the subset S^*, the separation problem in the dual can be formulated by defining the binary variables

$$
z_i := \begin{cases} 1, & \text{if object } i \text{ belongs to subset } S^* \\ 0, & \text{otherwise} \end{cases}
$$

and solving the system

$$
\sum_{i=1}^{m} y_i^* z_i > 1 \tag{73}
$$

$$
\sum_{i=1}^{m} d_i z_i \leq b \tag{74}
$$

$$
z_i \in \{0, 1\}, \quad i = 1, \ldots, m \tag{75}
$$

As in Section 2.1.1, the strict inequality (73) can be moved into the objective function

$$\max \sum_{i=1}^{m} y_i^* z_i \tag{76}$$

The overall ILP (76),(74),(75) is the Knapsack Problem with profits $(y_i^*), \forall i = 1, \ldots, n$

If the optimal solution $\overline{z}$ of the Knapsack has value ≤ 1, then *all* the dual constraints (72) are satisfied by y^*. Otherwise, given a subset $S^* := \{i \in \{1, \ldots, m\} : \overline{z}_i = 1\}$, any maximal set $S' \in \mathcal{S}$ such that $S^* \subseteq S'$ is the constraint (72) *most violated* by y^*

Example 13 Consider the Bin Packing problem with $n = 5$, $b = 10$, $d = (7, 5, 4, 4, 3)$

Starting from the (poor, for didactical purposes) initial feasible solution in which each object is packed separately into a (different) bin, and the associated variables for the LP (after having made the sets maximal) are x_S for $S \in \{\{1,5\},\{2,5\},\{3,5\},\{4,5\}\}$, the "column generation" algorithm evolves as

1. the nonzero (positive) components of $\tilde{x}^*$ and y^* are $\tilde{x}^*_S = 1$ for $S \in \{\{1,5\},\{2,5\},\{3,5\},\{4,5\}\}$ and $y_i^* = 1$ for $i \in \{1,2,3,4\}$; the constraint (72) associated with $S^* = \{2,3\}$ is violated and the corresponding variable $x_{\{2,3\}}$ is added
2. the nonzero (positive) components of $\tilde{x}^*$ and y^* are $\tilde{x}^*_S = 1$ for $S \in \{\{1,5\},\{2,3\},\{4,5\}\}$ and $y_i^* = 1$ for $i \in \{1,2,4\}$; the constraint (72) associated with $S^* = \{2,4\}$ is violated and the corresponding variable $x_{\{2,4\}}$ is added
3. the nonzero (positive) components of $\tilde{x}^*$ and y^* are $\tilde{x}^*_S = 1$ for $S \in \{\{1,5\},\{2,3\},\{4,5\}\}$ and $y_i^* = 1$ for $i \in \{1,3,4\}$; the constraint (72) associated with $S^* = \{3,4\}$ is violated and the corresponding variable $x_{\{3,4\}}$ is added
4. the nonzero (positive) components of $\tilde{x}^*$ and y^* are $\tilde{x}^*_S = 1$ for $\tilde{x}^*_{\{1,5\}} = 1$, $\tilde{x}^*_S = 1/2$ for $S \in \{\{2,3\},\{2,4\},\{3,4\}\}$, $y_1^* = 1$ and $y_i^* = 1/2$ for $i \in \{2,3,4\}$ and no constraint (72) is violated, thus $\tilde{x}^*$ and y^* are also the optimal primal and dual optimal solutions of the full LP, respectively

2.2.2 Vertex Coloring

The final example is the "column generation" algorithm for the solution of the continuous relaxation of the Vertex Coloring model (43)

The column generation problem to be solved at step 3 (*perfectly* equivalent to the separation problem of constraints (35) for the Stable Set) calls for finding constraints (44) violated by y^*, and can be formulated as follows

Given $y^* = (y_i^*) \geq 0$, find, if any, a subset $S^* \in \mathcal{S}$ such that $\sum_{i \in S^*} y_i^* > 1$

The problem can be solved by solving the maximum-weight Stable Set problem with weights (y_i^*), for example, formulated through the "intermediate" ILP model

Let us indicate with S^* the maximum-weight stable set with weights (y_i^*). If $\sum_{i \in S^*} y_i^* \leq 1$, then all constraints (44) are satisfied by y^*. Otherwise, any maximal stable set S' such that $S^* \subseteq S'$ defines the constraint (44) *most violated* by y^*

3 Applications

Among the large variety of routing problems associated with *Freight Transportation*, we restrict our attention to the most basic one, which, in turn, already includes the critical issue of having to deal with both classical routing (TSP) and loading (Bin Packing)

3.1 Capacitated Vehicle Routing Problem

Given a complete graph $G = (V, A)$ (the graph has been completed with arcs representing shortest paths in case of missing links), in the (Capacitated) Vehicle Routing problem we have

- a unique type of goods to be delivered (or collected)
- a unique depot, and, accordingly, we denote the set of nodes in the graph as $V = \{0, 1, \ldots, n\}$, where 0 is the node representing the depot, while $1, \ldots, n$ represent the clients
- a (scalar) demand d_j for each client j
- k *identical* vehicles with capacity b
- a cost c_a associated with each arc $a \in A$

We want to determine, for each vehicle, a *route* starting and ending at the depot such that

- each client is served by one vehicle, i.e., assuming the triangle inequalities (45) hold, each node but 0 is visited precisely once
- the sum of the demands of the clients served by the same vehicle does not exceed the capacity b
- the sum of the costs of the routes is minimized

Note that the problem is feasible if and only if the Bin Packing problem with n objects with weights (d_j) has a solution using k bins

In the next three sections we will discuss three different models for the problem. Those models have different characteristics and might be useful in different contexts because of the additional and heterogeneous constraints that could be added to the basic problem

3.1.1 (Classical) Two-index Formulation

The first model, similar to the ATSP one, uses a (binary) variable for each arc denoting if the arc belongs to some routes in the solution, without specifying to which one

$$x_a := \begin{cases} 1, & \text{if arc } a \text{ belongs to one of the } k \text{ routes} \\ 0, & \text{otherwise} \end{cases}$$

The "easy" part of the model has the form

$$\min \sum_{a \in A} c_a x_a \tag{77}$$

$$\sum_{a \in \delta^-(i)} x_a = 1, \quad i \in V \setminus \{0\} \tag{78}$$

$$\sum_{a \in \delta^+(i)} x_a = 1, \quad i \in V \setminus \{0\} \tag{79}$$

$$\sum_{a \in \delta^-(0)} x_a = k, \tag{80}$$

$$\sum_{a \in \delta^+(0)} x_a = k, \tag{81}$$

$$x_a \in \{0,1\}, \quad a \in A \tag{82}$$

and one needs to add constraints to avoid

- routes not touching the depot, and
- routes exceeding the capacity constraint

These two types of (forbidden) routes can be avoided by adding a single type of constraints, which are a complicated variant of the ATSP constraints (52)

For a given subset of clients $S \subseteq \{1,\ldots,n\}$, let us indicate with $\sigma(S)$ the value of the optimal solution of the Bin Packing problem with $|S|$ objects with weights given by the demands of the clients in S and capacity b, i.e., that of the vehicles

Then, the constraints to be added to make the model valid are

$$\sum_{a \in \delta^+(S)} x_a \geq \sigma(S), \quad S \subseteq V \setminus \{0\} \tag{83}$$

The separation of constraints (83) is trivial if the solution of the LP relaxation $x^* = (x^*_a)$ is integer (it is enough to check if the solution contains a route not touching the depot or exceeding the capacity), while it is extremely difficult if $x^* = (x_a^*)$ is fractional. More precisely, such a separation cannot be formulated with an (M)ILP, and only heuristic approaches have been developed

A weaker, but still sufficient to make the resulting model valid, variant of constraints (83) can be obtained by considering in the right-hand-side of the constraints the trivial lower bound value of the same Bin Packing problem (instead of its optimal solution value)

$$\sum_{a \in \delta^+(S)} x_a \geq \frac{\sum_{j \in S} d_j}{b}, \quad S \subseteq V \setminus \{0\} \tag{84}$$

The separation of constraints (84) is much easier. Given $x^* = (x_a^*)$, one wants to determine, if any, a subset $S^* \subseteq V \setminus \{0\}$ such that

$$\sum_{j \in S^*} d_j - b \sum_{a \in \delta^+(S^*)} x_a^* > 0$$

To formulate this separation problem as an ILP one needs the binary variables

$$y_i := \begin{cases} 1, & \text{if vertex } i \text{ belongs to } S^* \\ 0, & \text{otherwise} \end{cases}$$

and

$$z_a := \begin{cases} 1, & \text{if arc } a \text{ belongs to } \delta^+(S^*) \\ 0, & \text{otherwise} \end{cases}$$

Then, it exists a violated constraint (84) associated with S^* if and only if the optimal solution value of the following ILP is > 0

$$\max \sum_{i \in V} d_i y_i - b \sum_{a \in A} x_a^* z_a \tag{85}$$

$$y_0 = 0 \tag{86}$$

$$z_{(i,j)} \leq y_i, \quad (i,j) \in A \tag{87}$$

$$z_{(i,j)} \leq 1 - y_j, \quad (i,j) \in A \tag{88}$$

$$z_{(i,j)} \geq y_i - y_j, \quad (i,j) \in A \tag{89}$$

$$y_i, z_a \in \{0,1\}, \quad i \in V, \ a \in A \tag{90}$$

3.1.2 Three-index Formulation

Differently from the two-index formulation, the second model is straightforward to be adapted to the case in which the vehicles have different characteristics, for example in terms of capacity. Specifically, this model uses binary variables that specify if an arc is used by a specific vehicle, i.e., it belongs to a specific route

$$x_a^h := \begin{cases} 1, & \text{if arc } a \text{ belongs to route } h \\ 0, & \text{otherwise} \end{cases}$$

Moreover, we use binary variables to state if a client is served by a given route

$$y_i^h := \begin{cases} 1, & \text{if client } i \text{ is served by route } h \\ 0, & \text{otherwise} \end{cases}$$

The model is

$$\min \sum_{h=1}^{k} \sum_{a \in A} c_a x_a^h \tag{91}$$

$$\sum_{a \in \delta^-(i)} x_a^h = y_i^h, \quad h = 1, \ldots, k, \ i \in V \setminus \{0\} \tag{92}$$

$$\sum_{a \in \delta^+(i)} x_a^h = y_i^h, \quad h = 1, \ldots, k, \ i \in V \setminus \{0\} \tag{93}$$

$$\sum_{h=1}^{k} y_i^h = 1, \quad i \in V \setminus \{0\} \tag{94}$$

$$\sum_{a \in \delta^-(0)} x_a^h = 1, \quad h = 1, \ldots, k \tag{95}$$

$$\sum_{a \in \delta^+(0)} x_a^h = 1, \quad h = 1, \ldots, k \tag{96}$$

$$\sum_{i \in V \setminus \{0\}} d_i y_i^h \leq b, \quad h = 1, \ldots, k \tag{97}$$

$$\sum_{a \in \delta^+(S)} x_a^h \geq y_i^h, \quad h = 1, \ldots, k, \ S \subseteq V \setminus \{0\}, \ i \in S \tag{98}$$

$$x_a^h, y_i^h \in \{0, 1\}, \quad a \in A, \ i \in V \setminus \{0\} \tag{99}$$

(It is easy to observe that each variable y is defined by a sum of variables x, thus it would be possible to project variables y out but we keep them to improve readability of the model)

Constraints (97) trivially impose the capacity of each vehicle to be satisfied. Instead, constraints (98), which are a variant of the ATSP constraints (52), forbid subroutes not touching the depot by imposing that, if a client of a subset S is visited by a vehicle h, then at least one arc leaving S (i.e., going to clients outside S including the depot) must be selected

The separation of constraints (98) is similar to the separation of the ATSP constraints (52), with a structural difference

Given $\overline{x} = (\overline{x}_a^h)$ and $\overline{y} = (\overline{y}_i^h)$, the separation problem calls for determining, if any, a route h, a subset $\bar{S} \subseteq V \setminus \{0\}$ and a vertex $i \in \bar{S}$ such that

$$\sum_{a \in \delta^+(\bar{S})} \overline{x}_a^h < \overline{y}_i^h$$

In order to solve this separation problem, one has to consider all indices $h = 1, \ldots, k$, so as to find a violated constraint associated with a route h

Moreover, after h is fixed, one has to consider all clients i such that $\overline{y}_i^h > 0$, so as to find a violated constraint associated with route h and client i

Finally, after fixing both h and i, one has to find a subset of clients $\bar{S}$ (with $i \in \bar{S}$) by solving the ILP described in the following. This means that kn ILPs must be solved

By defining the binary variables

$$w_j := \begin{cases} 1, & \text{if client } j \text{ belongs to } \bar{S} \\ 0, & \text{otherwise} \end{cases}$$

and

$$z_a := \begin{cases} 1, & \text{if arc } a \text{ belongs to } \delta^+(\bar{S}) \\ 0, & \text{otherwise} \end{cases}$$

it exists a violated constraint associated with route h, client i and subset $\bar{S}$ if and only if the optimal solution value of the following ILP is $< \overline{y}_i^h$

$$\min \sum_{a \in A} \overline{x}_a^h z_a \tag{100}$$

$$w_0 = 0 \tag{101}$$

$$w_i = 1 \tag{102}$$

$$z_{(i,j)} \leq w_i, \quad (i,j) \in A \tag{103}$$

$$z_{(i,j)} \leq 1 - w_j, \quad (i,j) \in A \tag{104}$$

$$z_{(i,j)} \geq w_i - w_j, \quad (i,j) \in A \tag{105}$$

$$w_i, z_a \in \{0,1\}, \quad i \in V, \ a \in A \tag{106}$$

Although from the theoretical point of view, the three-index formulation does not add any benefit to the two-index one, it has considerably more flexibility. Not only constraints (97) can be adapted to the case in which each vehicle has a specific capacity b_h

$$\sum_{i \in V \setminus \{0\}} d_i y_i^h \leq b_h, \quad h = 1, \ldots, k$$

but also it is much easier to deal with the variant of the problem in which there are *time windows* imposing that each client i is visited in the time interval $[e_i, \ell_i]$.

We deal with the time window variant by assuming that

- each vehicle starts at the depot at time 0
- the travel time on any arc a is equal to the cost of the arc c_a
- the delivery time is null
- no vehicle can either slow down or stop so as the travel time of the route is bigger than its cost

Under those assumptions, the time window constraints can be imposed by defining variables

$$s_i^h := \begin{cases} \text{time at which client } i \text{ is served by vehicle/route } h \text{ if } y_i^h = 1 \\ 0 \text{ otherwise} \end{cases}$$

and the constraints

$$\begin{aligned} s_i^h &\geq e_i y_i^h, \quad h = 1, \ldots, k, \ i \in V \setminus \{0\} && (107) \\ s_i^h &\leq \ell_i y_i^h, \quad h = 1, \ldots, k, \ i \in V \setminus \{0\} && (108) \\ s_0^h &= 0, \quad h = 1, \ldots, k && (109) \\ x_{(i,j)}^h = 1 &\Rightarrow s_j^h = s_i^h + c_{(i,j)}, \quad h = 1, \ldots, k, \ i \in V, \ j \in V \setminus \{0\} && (110) \end{aligned}$$

In order to transform the logical constraints (110) into linear ones, one needs to impose "bigM" constraints like those used for the Fixed-Charge problem. More precisely, the imposed constraints must be deactivated if $x_{(i,j)}^h = 0$, i.e., if j does not immediately follow i in route h. This is obtained with

$$\begin{aligned} s_j^h &\geq s_i^h + c_{(i,j)} - M(1 - x_{(i,j)}^h), \quad h = 1, \ldots, k, \ i \in V, \ j \in V \setminus \{0\} && (111) \\ s_j^h &\leq s_i^h + c_{(i,j)} + M(1 - x_{(i,j)}^h), \quad h = 1, \ldots, k, \ i \in V, \ j \in V \setminus \{0\} && (112) \end{aligned}$$

It is very easy to construct examples in which the continuous relaxation of the above model is very "weak", that is, where very small (fractional) values of variables $x^h_{(i,j)}$ are enough to deactivate the constraints

In practice, it would be good to avoid "bigM" of this type, or whenever that turns out to be impossible, use relatively small (but still "safe") values of M and, in any case, managing the models with a lot of care recognizing the intrinsic weakness of their LP relaxation

3.1.3 Set Partitioning Formulation

The last model we consider for the Vehicle Routing is a Set Partitioning-type model similar to that introduced for the Bin Packing and the Vertex Coloring problems. It is mostly used when complex constraints defining the feasibility of the routes have to be considered

Let us define $\mathcal{C}$ to be the collection of all *feasible* routes and let us call γ_C the cost of route $C \in \mathcal{C}$. Moreover, let us call $\mathcal{C}_i$ the subcollection of the routes of $\mathcal{C}$ that serve client i. By defining the binary variables

$$x_C := \begin{cases} 1, & \text{if the solution uses route C} \\ 0, & \text{otherwise} \end{cases}$$

one can define the following ILP formulation

$$\min \sum_{C \in \mathcal{C}} \gamma_C x_C \tag{113}$$

$$\sum_{C \in \mathcal{C}} x_C = k, \tag{114}$$

$$\sum_{C \in \mathcal{C}_i} x_C = 1, \quad i \in V \setminus \{0\} \tag{115}$$

$$x_C \in \{0,1\}, \quad C \in \mathcal{C} \tag{116}$$

Because we assume triangle inequalities hold, then constraints (115) could potentially be written in "$\geq$" form as well

Note that for the collection $\mathcal{C}$ we on purpose avoided restricting ourselves to *maximal* routes, i.e., routes that cannot serve additional clients because of the capacity limitation. The reason is that the cost γ_C of any route C depends on the route itself, i.e., on the sequence of the clients and not only on the set of the clients. This is different from the Bin Packing and Vertex Coloring, where it is always possible to remove an "object" (item or vertex, respectively) from a "set" (bin or stable set, respectively) *without* changing the cost of the set

The same holds, in general, for all models in which the cost of the variables is not equal. For them, it is not correct to limit the variable definition to those associated with maximal sets

The column generation problem associated with the continuous relaxation of the above model (obtained by dropping the integrality constraint on the x variables and replace it by the requirement to be nonnegative) requires to write down the corresponding dual

Let us call I_C the set of clients served by route $C \in \mathcal{C}$ and let us consider the variable z associated with equation (114) and variables y_i associated with equations (115). Note that because constraints (114) and (115) are in equality form the associated dual variables are free

Then, the dual reads as follows

$$\max kz + \sum_{i \in V \setminus \{0\}} y_i \tag{117}$$

$$z + \sum_{i \in I_C} y_i \leq \gamma_C, \quad C \in \mathcal{C} \tag{118}$$

Given z^* and $y^* = (y_i^*)$ (solution of the dual of the restricted primal), the column generation problem calls for determining, if any, a route $C^* \in \mathcal{C}$ such that

$$z^* + \sum_{i \in I_{C^*}} y_i^* > \gamma_C^* \Leftrightarrow \sum_{i \in I_{C^*}} y_i^* - \sum_{a \in C^*} c_a > -z^*$$

Such a route C^* exists if and only if the optimal solution value of the ILP described in the following is $> -z^*$ (and in that case the ILP solution gives it)

By defining the binary variables

$$w_i := \begin{cases} 1, & \text{if client } i \text{ is served by route } C^*, \text{ i.e., } i \in I_{C^*} \\ 0, & \text{otherwise} \end{cases}$$

and

$$u_a := \begin{cases} 1, & \text{if arc } a \text{ belongs to } C^* \\ 0, & \text{otherwise} \end{cases}$$

then the column generation model (analogous to the three-index formulation but written for a single vehicle/route) is

$$\max \sum_{i\in V\setminus\{0\}} y_i^* w_i - \sum_{a\in A} c_a u_a \tag{119}$$

$$\sum_{a\in\delta^-(i)} u_a = w_i, \quad i \in V \setminus \{0\} \tag{120}$$

$$\sum_{a\in\delta^+(i)} u_a = w_i, \quad i \in V \setminus \{0\} \tag{121}$$

$$\sum_{a\in\delta^-(0)} u_a = 1, \tag{122}$$

$$\sum_{a\in\delta^+(0)} u_a = 1, \tag{123}$$

$$\sum_{i\in V\setminus\{0\}} d_i w_i \leq b \tag{124}$$

$$\sum_{a\in\delta^+(S)} u_a \geq w_i, \quad S \subseteq V \setminus \{0\},\ i \in S \tag{125}$$

$$u_a, w_i \in \{0,1\}, \quad a \in A,\ i \in V \setminus \{0\} \tag{126}$$